

TUGAS AKHIR
PEMBUATAN APLIKASI MONITORING PROSES KLAIM ASURANSI
PT. ASURANSI ASTRA BUANA SURAKARTA

Diajukan Untuk Memenuhi Salah Satu Syarat Mencapai Gelar Ahli Madya
Program Diploma III Ilmu Komputer



Diajukan Oleh:

Arum Rahmawati

M.3206009

PROGRAM D-III ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SEBELAS MARET
SURAKARTA
2009

HALAMAN PERSETUJUAN

**PEMBUATAN APLIKASI MONITORING PROSES KLAIM ASURANSI
PT. ASURANSI ASTRA BUANA SURAKARTA**

Disusun Oleh

ARUM RAHMAWATI
NIM. M03202002

Tugas Akhir ini telah disetujui untuk dipertahankan
di hadapan dewan penguji
pada tanggal _____

Di bimbing oleh
Pembimbing Utama

Irwan Susanto. S.Si, DEA
NIP. 19710511 199512 1001

HALAMAN PENGESAHAN

PEMBUATAN APLIKASI MONITORING PROSES KLAIM ASURANSI PT. ASURANSI ASTRA BUANA SURAKARTA

Disusun Oleh

ARUM RAHMAWATI
NIM. M3206009

Di bimbing oleh
Pembimbing Utama

Irwan Susanto. S.Si, DEA
NIP. 19710511 199512 1001

Tugas Akhir ini telah diterima dan disahkan
oleh dewan penguji Tugas Akhir
Program Diploma III Ilmu Komputer
pada hari _____ tanggal _____

Dewan Penguji

1. Penguji 1 Bowo Winarno, S.Si, M.Kom ()
NIP. 19810430 200812 1001
2. Penguji 2 Mania Rositha, M.Si ()
NIP. 19520628 198303 2001

Disahkan Oleh

Dekan
Fakultas MIPA UNS

Ketua Program Studi
DIII Ilmu Komputer UNS

Prof. Drs. Sutarno, M.Sc.Ph.D
NIP. 19600809 198612 1001

Drs. Y.S Palgunadi, M.Sc
NIP. 19560407 198303 1004

ABSTRACT

Arum Rahmawati. 2009. THE MAKING OF INSURANCE CLAIM PROCESS MONITORING APPLICATION IN PT. ASURANSI ASTRA BUANA SURAKARTA. D3 Computer Program, Mathematics and Science Faculty, Surakarta Sebelas Maret University.

The use of database really helps in the data monitoring process such as monitoring the insurance claim process. PT. Asuransi Astra Buana Surakarta still uses the Microsoft Excel program as the processor of its insurance claim process monitoring. Microsoft Excel is slightly slower in processing data to get new information. For that reason, this research aims to develop a special program functioning to do the insurance claim monitoring job in PT. Asuransi Astra Buana Surakarta.

Software Borland Delphi 7.0 was used to develop an Application of Insurance Claim Process Monitoring on PT. Asuransi Astra Buana Surakarta. Database MySQL was used for storing the data after being processed. Meanwhile the ODBC (Open Database Connectivity) connector 5.15-win 32 was used to connect the Borland Delphi 7.0 application to the database MySQL. In this study, the writer used two research techniques: primary data and secondary data collection ones. The technique of collecting primary data consists of three methods: interview, observation and documentation. For the technique of collecting secondary data, the writer used literature study. The application of Insurance Claim Process Monitoring made with Borland Delphi 7.0 aid is more effective than that with Microsoft Excel. This can be seen from the faster data processing. This application is also more efficient because it facilitate the users in monitoring data.

INTI SARI

Arum Rahmawati. 2009. PEMBUATAN APLIKASI MONITORING PROSES KLAIM ASURANSI PT. ASURANSI ASTRA BUANA SURAKARTA. Program D3 Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sebelas Maret Surakarta.

Penggunaan *database* sangat membantu dalam proses monitoring data seperti monitoring proses klaim asuransi. PT. Asuransi Astra Buana Surakarta masih menggunakan program *Microsoft Excel* sebagai pengolah monitoring proses klaim asuransinya. *Microsoft Excel* sedikit lambat dalam pengolahan data untuk memperoleh informasi baru. Oleh karena itu, penelitian ini bertujuan untuk membuat program khusus yang berfungsi untuk melakukan *monitoring* klaim asuransi pada PT. Asuransi Astra Buana Surakarta.

Software Borland Delphi 7.0 digunakan untuk membuat Aplikasi Monitoring Proses Klaim Asuransi PT. Asuransi Astra Buana Surakarta. *Database MySQL* digunakan untuk menyimpan data setelah diproses. Sedangkan konektor ODBC (*Open Database Connectivity*) 5.15- win32 digunakan untuk menghubungkan aplikasi Borland Delphi 7.0 ke *database MySQL*. Dalam penelitian ini, penulis menggunakan dua teknik penelitian yaitu teknik pengumpulan data primer dan data sekunder. Untuk teknik pengumpulan data primer terdiri dari tiga metode yaitu metode *interview*, metode observasi dan metode dokumentasi. Untuk teknik pengumpulan data sekunder penulis melakukan studi literatur. Aplikasi Monitoring Proses Klaim Asuransi yang dibuat dengan bantuan Borland Delphi 7.0 lebih efektif dibanding menggunakan *Microsoft Excel*. Hal ini terlihat dalam proses pengolahan data yang lebih cepat. Aplikasi ini juga lebih efisien karena memudahkan pengguna dalam proses monitoring data.

MOTTO

Semangat juang yang tinggi menghantarkan potensi diri dalam berprestasi.

Tetap semangat dan terus semangat. OK

(Penulis)

PERSEMBAHAN

Kupersembahkan buah tangan ini untuk:

- 1. Kedua Orang Tua Terkasih dan Tercinta*
- 2. Mas Arif Santoso Tersayang*
- 3. Dek Arinda dan Mbak Anis Tersayang*

KATA PENGANTAR

Puji syukur kupanjatkan atas karunia Allah SWT sehingga penulis dapat menyelesaikan Laporan Tugas Akhir dengan judul "PEMBUATAN APLIKASI MONITORING PROSES KLAIM ASURANSI PT. ASURANSI ASTRA BUANA SURAKARTA" dengan baik. Tugas akhir ini merupakan salah satu syarat yang harus dipenuhi guna menyelesaikan studi di program studi Diploma 3 Ilmu Komputer FMIPA Universitas Sebelas Maret Surakarta.

Dalam penyelesaian laporan ini, penulis dibantu oleh berbagai pihak. Untuk itu, penulis menghaturkan banyak terima kasih kepada:

- 1) Bapak Irwan Susanto, S.Si, DEA, selaku dosen pembimbing Tugas Akhir.
- 2) Drs. YS. Palgunadi, M.Sc selaku ketua program studi D3 Ilmu Komputer Universitas Sebelas Maret Surakarta.
- 3) Bapak Haris Bachtiar selaku manager PT. Asuransi Astra Buana Surakarta yang telah memberikan waktu dan tempat untuk melaksanakan penelitian.
- 4) Ayah dan Ibu yang selalu memberi doa, dukungan dan nasehat.
- 5) Serta semua pihak yang telah membantu dalam proses penyelesaian penyusunan laporan tugas akhir ini.

Penulis menyadari bahwa suatu karya dalam bidang apapun tidak terlepas dari kekurangan, maka penulis mengharapkan kritik dan saran yang membangun. Semoga laporan tugas akhir ini bermanfaat.

Surakarta, Juni 2009

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERSETUJUAN	ii
HALAMAN PENGESAHAN	iii
ABSTRACT	iv
INTISARI	v
MOTTO	vi
PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	1
1.3 Batasan Masalah	2
1.4 Tujuan dan Manfaat	2
1.5 Metodologi Penelitian	3
1.6 Sistematika Penulisan	3
BAB II LANDASAN TEORI	5
2.1 Pengertian Data	5
2.2 Pengertian Sistem	5
2.3 Pengertian Informasi	6
2.4 Pengertian Sistem Informasi	6
2.5 Perancangan <i>Database</i>	7
2.6 Pengertian Perancangan Sistem	8
2.7 <i>Borland Delphi 7.0</i>	10
2.8 DBMS	11
2.9 <i>MySQL</i>	12
2.10 ODBC	12
BAB III DESAIN DAN PERANCANGAN	13

3.1 Desain Aplikasi	13
3.1.1 <i>Flowchart</i>	13
3.1.2 <i>Context Diagram</i> (CD)	14
3.1.3 <i>Data Flow Diagram</i> (DFD)	15
3.1.4 <i>Data Dictionary</i> (DD)	17
3.1.5 Struktur Data	20
3.1.6 <i>Model Relasional Database</i> (RDBMS)	24
3.1.7 Entity Relationship Diagram	25
3.2 Perancangan Aplikasi	25
3.2.1 Kebutuhan <i>Hardware</i> dan <i>Software</i>	26
BAB IV IMPLEMENTASI DAN ANALISA.....	27
4.1 Implementasi Program	27
4.1.1 Desain <i>Input</i>	27
4.1.2 Desain <i>Output</i>	33
4.2 Analisa Program	36
4.2.1 Keunggulan Program	36
4.2.2 Kelemahan Program	36
BAB V PENUTUP	37
5.1 Kesimpulan	37
5.2 Saran	37
DAFTAR PUSTAKA	38

DAFTAR TABEL

	Halaman
Tabel 2.1 Simbol-simbol <i>Context Diagram</i>	8
Tabel 2.2 Simbol-simbol <i>Data Flow Diagram</i>	9
Tabel 2.3 Simbol-simbol <i>Entity Relationship Diagram</i>	10
Tabel 3.1 Tabel <i>Customer</i>	20
Tabel 3.2 Tabel Kendaraan	21
Tabel 3.3 Tabel Polis Tertanggung	21
Tabel 3.4 Tabel <i>Survey</i>	22
Tabel 3.5 Tabel Bengkel	22
Tabel 3.6 Tabel Klaim	23
Tabel 3.7 Tabel ACC diTolak	23

DAFTAR GAMBAR

	Halaman
Gambar 3.1 <i>Flowchart</i>	13
Gambar 3.2 <i>Context Diagram</i>	14
Gambar 3.3 DFD <i>Level 1</i>	15
Gambar 3.4 DFD <i>Level 2</i> Proses 1	16
Gambar 3.5 DFD <i>Level 2</i> Proses 2	16
Gambar 3.6 <i>Model Relational Database (RDBMS)</i>	24
Gambar 3.7 <i>Entity Relationship Diagram (ERD)</i>	25
Gambar 4.1 <i>Form Utama</i>	27
Gambar 4.2 <i>Form Data Customer</i>	28
Gambar 4.3 <i>Form Data Bengkel</i>	29
Gambar 4.4 <i>Form Pengajuan Klaim</i>	30
Gambar 4.5 <i>Form Klaim ACC/ Gagal ACC</i>	31
Gambar 4.6 <i>Form Survey</i>	32
Gambar 4.7 <i>Form Monitoring Survey</i>	33
Gambar 4.8 Laporan Data <i>Customer</i>	34
Gambar 4.9 Laporan Data Bengkel	34
Gambar 4.10 Laporan Grafik <i>Loading</i> Bengkel	35
Gambar 4.12 Laporan Klaim ACC dan Gagal ACC	35

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi komputer, memiliki peranan penting dalam menyimpan dan mendapatkan kembali data secara efektif dan efisien. Sistem yang sering dipakai adalah *database system* dan *database management system* (DBMS) yang merupakan *software* yang digunakan untuk mengatur jalannya *input* dan *output* data. Salah satu fasilitas dari DBMS adalah dapat mengakses sebuah *database* tunggal secara bersamaan oleh banyak *user*, dapat mengakses data secara terbatas hanya untuk *user* yang berhak dan mengganti kegagalan dari sistem tanpa kehilangan keutuhan data.

Dalam penggunaannya, *database* sangat membantu dalam proses monitoring data, adapun contohnya adalah monitoring proses klaim asuransi yang diterapkan oleh PT. Asuransi Astra Buana Surakarta. Badan usaha ini, dalam melakukan pengolahan monitoring proses klaim asuransi masih menggunakan program *microsoft excel* sehingga proses pengolahan data sedikit lambat dalam memperoleh informasi baru. Untuk itu, perlu dibuat program khusus yang berfungsi untuk melakukan *monitoring* klaim asuransi pada PT. Asuransi Astra Buana Surakarta.

1.2 Rumusan Masalah

Dari uraian latar belakang di atas, maka penulis dapat memperoleh rumusan masalah yang akan menjadi pokok bahasan yaitu sebagai berikut: Bagaimana membuat aplikasi monitoring proses klaim asuransi PT. Asuransi Astra Buana Surakarta dengan menggunakan *software* Borland Delphi 7.0?

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini adalah membuat program klaim asuransi yang hanya meliputi:

1. Aplikasi monitoring proses klaim asuransi di PT. Asuransi Astra Buana Surakarta.
2. Aplikasi ini dijalankan pada satu buah komputer yang bersifat *stand alone*.
3. Monitoring *survey*.

1.4 Tujuan dan Manfaat

1.4.1 Tujuan dari penyusunan tugas akhir ini adalah:

Penulis dapat membuat sebuah program aplikasi monitoring dengan menggunakan Borland Delphi 7.0 untuk proses klaim asuransi PT. Asuransi Astra Buana Surakarta.

1.4.2 Penyusunan tugas akhir ini diharapkan dapat memberikan manfaat sebagai berikut:

a) Bagi Penulis

1. Sebagai dasar untuk mengetahui hal-hal yang berhubungan dengan analisis proses klaim asuransi, sehingga diharapkan dengan penyusunan tugas akhir ini, akan mampu membimbing penulis menjadi seorang analisis.
2. Sebagai bahan referensi khususnya bagi rekan-rekan mahasiswa dalam melakukan pengembangan penelitian dan menambah pengetahuan.
3. Penulis dapat menghasilkan sebuah aplikasi monitoring klaim asuransi untuk instansi tersebut dan hasil aplikasi tersebut dapat dimanfaatkan.

b) Bagi Instansi

1. Memperoleh sebuah karya baru yang dapat digunakan untuk meningkatkan mutu kinerja pada instansi tersebut.
2. Sebagai wadah eksperimen bagi perusahaan dimana kesimpulan dan saran yang diberikan oleh penulis dapat dijadikan sebagai bahan pertimbangan khususnya dalam pengambilan keputusan.

1.5 Metodologi Penelitian

Dalam penelitian ini, penulis menggunakan dua teknik pengumpulan data antara lain sebagai berikut:

a) Teknik pengumpulan data primer.

1. Metode *Interview*

Pengumpulan data dengan mengajukan beberapa pertanyaan kepada pimpinan PT. Asuransi Astra Buana Surakarta serta staf yang terkait pada bidangnya.

2. Metode Observasi

Teknik pengumpulan data dengan cara melakukan pengamatan dan penelitian secara langsung khususnya terhadap transaksi-transaksi yang terjadi pada kegiatan operasional proses klaim asuransi.

3. Metode Dokumentasi

Metode dokumentasi dilakukan dengan cara mengumpulkan data dari bagian yang berhubungan dengan pengarsipan.

b) Teknik pengumpulan data sekunder

Teknik pengumpulan data ini biasa disebut dengan metode studi literatur diperoleh penulis dengan cara melakukan penelitian kepustakaan yang menjadi pendukung dan pelengkap penelitian lapangan.

1.6 Sistematika Penulisan

Untuk memberikan gambaran tentang pembahasan tugas akhir ini, maka secara garis besar pembahasan studi akhir ini adalah sebagai berikut:

BAB I : Pendahuluan meliputi latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian serta metodologi penelitian.

BAB II : Landasan teori meliputi dasar-dasar teoritis yang menjadi landasan pemecahan masalah.

BAB III : Desain dan perancangan memuat tentang data-data yang diperlukan dalam perancangan pembuatan aplikasi seperti *flowchart*, *context diagram*, *data flow diagram (DFD)*, *data dictionary*, struktur data dan *model relasional*.

BAB IV : Pembahasan berisikan tentang implementasi dan analisa yang merupakan langkah dan hasil analisa penelitian meliputi *form input*, *form output* serta pembahasan yang sifatnya terpadu.

BAB V : Penutup memuat kesimpulan dan saran yang berisi tentang pernyataan singkat dan tepat yang dijabarkan dari hasil penelitian serta memuat saran yang berisi tentang sumbang saran pemikiran yang didasarkan pada kesimpulan yang diperoleh untuk penyempurnaan dan pengembangan dimasa yang akan datang.

BAB II

LANDASAN TEORI

Dalam pembuatan tugas akhir ini, diperlukan beberapa definisi dan pengertian yang berhubungan dengan penyelesaian tugas akhir. Beberapa hal tersebut diantaranya: data, sistem, informasi, sistem informasi, *database*, perancangan sistem informasi, Borland Delphi 7.0, DBMS, *MySQL* dan ODBC.

2.1 Pengertian Data

Data merupakan bahan keterangan tentang kejadian-kejadian nyata atau fakta-fakta yang dirumuskan dalam sekelompok lambang tertentu yang tidak acak yang menunjukkan jumlah, tindakan atau hal. Data dapat berupa catatan-catatan dalam kertas, buku atau tersimpan sebagai *file* dalam basis data (Edhy Sutanta, 2004).

Data adalah fakta mengenai objek, orang dan lain-lain. Data dinyatakan dengan nilai angka, deretan karakter atau simbol. Data didalam basis data mempunyai sifat terpadu (*integrated*) dan berbagi (*shared*). Tepadu berarti bahwa berkas-berkas data yang ada pada basis data saling terkait, tetapi kemubaziran data tidak akan terjadi sedikit sekali. Berbagi data berarti bahwa data dapat dipakai oleh sejumlah pengguna lebih tegas lagi sesuatu data dapat diakses oleh sejumlah pengguna dalam waktu bersamaan (Abdul Kadir, 2001).

2.2 Pengertian Sistem

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu (Jogiyanto, 2001).

2.3 Pengertian Informasi

Informasi adalah suatu bentuk penyajian data yang melalui mekanisme pemrosesan, yang berguna bagi pihak tertentu misalnya *manager*. Bagi pihak manajemen, informasi merupakan bahan untuk pengambil keputusan (Abdul Kadir, 2003).

Informasi adalah data yang digunakan dalam pengambilan keputusan alasannya adalah bahwa informasi masih bersifat relatif, relatif terhadap situasi, relatif terhadap waktu saat keputusan diambil, juga relatif terhadap pembuatan keputusan dan bahkan juga terhadap latar belakang pengambil keputusan (Abdul Kadir, 2001).

Informasi merupakan hasil pengolahan data sehingga menjadi bentuk yang penting bagi penerimanya dan mempunyai kegunaan sebagai dasar dalam pengambilan keputusan yang dapat dirasakan akibatnya secara langsung saat itu juga atau tidak secara langsung pada saat mendatang (Edhy Sutanta, 2004).

Informasi merupakan hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan (Faried Irmansyah, 2003).

2.4 Pengertian Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan lapisan-lapisan yang diperlukan (Jogiyanto, 2001).

Sistem informasi adalah suatu sistem dimana suatu organisasi yang merupakan kombinasi dari orang, fasilitas, teknologi, media, prosedur dan pengendalian untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya

terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan (Faried Irmansyah, 2003).

2.5 Perancangan Database

Database adalah kumpulan *file-file* yang saling berelasi, relasi tersebut biasa ditunjukkan dengan kunci dari tiap *file* yang ada. *Database* merupakan kumpulan *file-file* yang mempunyai kaitan antara satu *file* dengan *file* yang lain sehingga membentuk satu bangunan data untuk menginformasikan satu perusahaan, instansi dalam batasan tertentu. Bila terdapat *file* yang tidak dapat dipadukan atau dihubungkan dengan *file* yang lainnya berarti *file* tersebut bukanlah kelompok dari satu *database*, ia akan dapat membentuk satu *database* sendiri (Harianto Kristanto, 1994).

Basis data (*database*) adalah kumpulan dari berbagai data yang saling berhubungan satu dengan yang lainnya. Basis data tersimpan di perangkat keras serta dimanipulasi dengan menggunakan perangkat lunak. Pendefinisian basis data meliputi spesifikasi dari tipe data, struktur dan batasan dari data atau informasi yang akan disimpan. *Database* merupakan salah satu komponen yang penting dalam sistem informasi karena merupakan basis dalam menyediakan informasi pada para pengguna atau *user* (Krida, 2009).

Database dapat diungkapkan sebagai suatu pengorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat. Dalam hal ini, pengertian akses dapat mencakup pemerolehan data maupun pemanipulasian data seperti menambah dan menghapus data (Abdul Kadir, 2003).

Database merupakan kumpulan dari item yang saling berhubungan satu dengan yang lainnya yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, tersimpan di *hardware* komputer dan dengan *software* untuk melakukan manipulasi untuk kegunaan tertentu. *Database* merupakan kumpulan dari *file* atau tabel membentuk suatu *database* (Faried Irmansyah, 2003).

2.6 Pengertian Perancangan Sistem

Analisis perancangan sistem adalah suatu pendekatan yang sistematis untuk mengidentifikasi masalah, peluang, dan tujuan-tujuan, menganalisis arus informasi dalam organisasi, serta untuk merancang sistem informasi terkomputerisasi untuk menyelesaikan suatu masalah. Saat informasi berkembang, suatu pendekatan yang sistematis dan terencana untuk memperkenalkan, memodifikasi, dan pemeliharaan sistem informasi menjadi sangat penting. Analisis dan perancangan sistem menampilkan pendekatan semacam itu (Kendall & Kendall, 2003).

Dalam suatu perancangan sistem diperlukan alat bantu untuk mempermudah dalam menganalisis sebuah sistem informasi. Alat bantu yang digunakan dalam menganalisis sistem informasi tersebut antara lain:



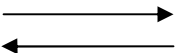
1. *Flowchart*

Flowchart merupakan alat bantu untuk membuat program dan untuk kepentingan dokumentasi. Alat bantu jenis ini berupa grafik atau simbol-simbol yang menggambarkan kegiatan-kegiatan yang ada pada sebuah program.

2. *Context Diagram* (CD)

Context Diagram merupakan diagram yang paling tidak detail dari sistem informasi yang menggambarkan alur data kedalam dan keluar sistem. Simbol-simbol yang digunakan dalam *context diagram* adalah sebagai berikut:



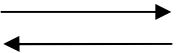


Tabel 2.1 Simbol – simbol *Context Diagram*

Simbol	Keterangan
	Menggambarkan proses dimana aliran data ditransformasikan ke aliran data keluar.
	Simbol entitas eksternal yang menggambarkan asal atau tujuan data suatu proses.
	Menggambarkan sistem alir data atau aliran data.

3. *Data Flow Diagram* (DFD)

Data Flow Diagram merupakan gambaran keseluruhan kerja sistem secara garis besar. Merupakan representasi dari sistem DFD menggunakan komponen dari sebuah sistem, alir data diantar ke komponen-komponen asal, tujuan dan penyimpanan dari data tersebut. Simbol-simbol DFD yang digunakan dalam menggambarkan kerja sistem adalah sebagai berikut:

Tabel 2.2 Simbol –simbol *Data Flow Diagram*

Simbol	Keterangan
	Menunjukkan kesatuan luar atau eksternal yang bisa menerima informasi atau data-data awal.
	Menunjukkan proses dimana beberapa tindakan atau sekelompok tindakan dijalankan
	Menunjukkan arus data dimana informasi sedang melintas menuju kesuatu proses.
	<i>File</i> basis data atau penyimpanan yang diimplementasikan dalam komputer.
	<i>Report generator</i> yang merupakan dokumen yang tersimpan dalam jumlah besar.

4. *Data Dictionary* (DD)

Data Dictionary atau kamus data adalah katalog fakta tentang data dan kebutuhan informasi, dengan kamus data analisis sistem dapat didefinisikan data yang mengalir kesistem dengan lengkap. Kamus data ini memiliki fungsi untuk menjelaskan arti aliran data dan penyimpanan dalam penggambaran DFD, mendeskripsikan komposisi paket data serta menjelaskan spesifikasi nilai dan satuan yang *relevant* terhadap data yang mengalir dalam sistem.

5. *Model Relasional Database* (RDBMS)

DBMS adalah sistem pengorganisasian dan pengolahan *database* pada komputer. Sistem ini dirancang untuk mampu melakukan berbagai data


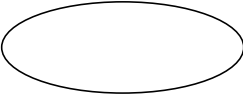
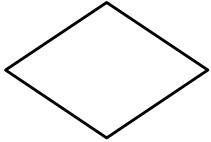

dengan beberapa referensi data yang sama. DBMS ini mampu diakses oleh berbagai aplikasi. Terobosan dari DBMS adalah *Relasional Database Management System* (RDBMS) yang mengorganisasikan data dalam suatu struktur dan memaksimalkan berbagai cara serta menghubungkan antar kumpulan data yang disimpan dalam *database*.

Terobosan berikutnya adalah *Distributed Relasional Database Management System* (DRDBMS). Dengan DRDBMS memungkinkan informasi berada pada baris data di lokasi yang berbeda (didistribusikan) dan direferensikan, diperbaharui, dan diakses dari semua lokasi seolah-olah data tersebut berbasis tunggal dan terpusat.

6. Entity Relationship Diagram (ERD)

Entity Relationship Diagram adalah suatu model konseptual yang mendeskripsikan hubungan antara penyimpanan data (file data). ERD digunakan untuk memodelkan struktur data dan hubungan antar data serta untuk menguji model dengan menyampaikan proses yang harus dilakukan (Fatansyah, 2004).

Tabel 2.3 Simbol –simbol *Entity Relationship Diagram*

Simbol	Keterangan
	Entitas digunakan untuk menggambarkan obyek yang dapat diidentifikasi dalam lingkungan pemakai.
	Atribut digunakan untuk menggambarkan elemen-elemen dari suatu entitas yang menggambarkan karakter entitas.
	Merupakan hubungan entitas satu dengan entitas yang lain. Hubungan tersebut dinamakan dengan relasi.
	Garis digunakan untuk menghubungkan entitas dan entitas dengan atribut.

2.7 Borland Delphi 7.0

Bahasa *Delphi* merupakan bahasa pemrograman yang dapat diatur sedemikian rupa sehingga dapat bekerja sama dengan piranti lain sehingga

membentuk suatu sistem pengendali. Bahasa *delphi* mempunyai *form* dan *editor* program untuk membuat program. Dengan menggabungkan *form* dan *editor* program, *delphi* dapat digunakan sebagai *software* yang berisi urutan perintah untuk mengendalikan suatu alat terkendali (Muhammad Supriadi, 2005)

Khusus untuk pemrograman *database*, *Borland Delphi* menyediakan fasilitas objek yang kuat dan lengkap yang memudahkan programmer dalam membuat program. Format *database* yang dimiliki *Delphi* adalah format *database Paradox, dBase, MS. Access, ODBC, SyBase, Oracle, MySQL* dan lain-lain.

2.8 DBMS

DBMS singkatan dari *Database Management System*. *DBMS* merupakan perangkat lunak atau program komputer yang dirancang secara khusus untuk memudahkan pengelolaan *database*. Salah satu macam *DBMS* yang populer dewasa ini berupa *RDBMS (Relational DataBase Management System)*, yang menggunakan model basis data relasional atau dalam bentuk tabel-tabel yang saling terhubung (Abdul Kadir, 2003).

Sebuah sistem pengelola basis data (*Database Management System* atau *DBMS*) terbagi atas modul-modul yang masing-masing memiliki tanggung jawab dalam membentuk struktur sistem basis data secara keseluruhan. *DBMS* sederhana seperti *dBase, Paradox* atau *MS-Access* misalnya, tidak mengakomodasi pemakaian basis data oleh banyak pemakai. Tetapi dengan *DBMS* semacam *Oracle, Sybase, Ingres* atau *My-SQL server* hal itu telah terakomodasi dengan baik (Fatansyah, 2004).

2.9 MySQL

SQL merupakan bahasa *query* yang paling banyak dipilih oleh *DBMS* dan *Development Tolls* (seperti *Visual Basic, Delphi, Power Builder, Java, dll*) dalam menyediakan median bagi penggunaanya untuk berinteraksi dengan basis data (Fatansyah, 2004).

MySQL merupakan salah satu contoh produk *RDBMS* yang sangat populer di lingkungan *linux*, tetapi juga tersedia pada *windows*. Banyak situs *web* yang

menggunakan *MySQL* sebagai *database server* (*server* yang melayani permintaan akses terhadap *database*). *MySQL* sebagai *database server* juga dapat diakses melalui program yang dibuat dengan menggunakan *Borland Delphi* (Abdul Kadir, 2003).

2.10 ODBC

ODBC (*Open Database Connectivity*) merupakan antar muka pemrograman standar industri yang memungkinkan aplikasi-aplikasi mengakses berbagai *system* manajemen basis data yang berada dalam berbagai *platform*. *ODBC* memungkinkan ketidaktergantungan basis data melalui kaidah *SQL* yang standar. Perangkat lunak *ODBC* tersusun atas sejumlah komponen, yaitu lapisan aplikasi, manajer pengendali (*driver manager*), pengendali yang spesifik terhadap basis data, dan sumber data. *ODBC* ini dibutuhkan untuk koneksi aplikasi ke dalam *database MySQL* (Abdul Kadir, 2003).

BAB III

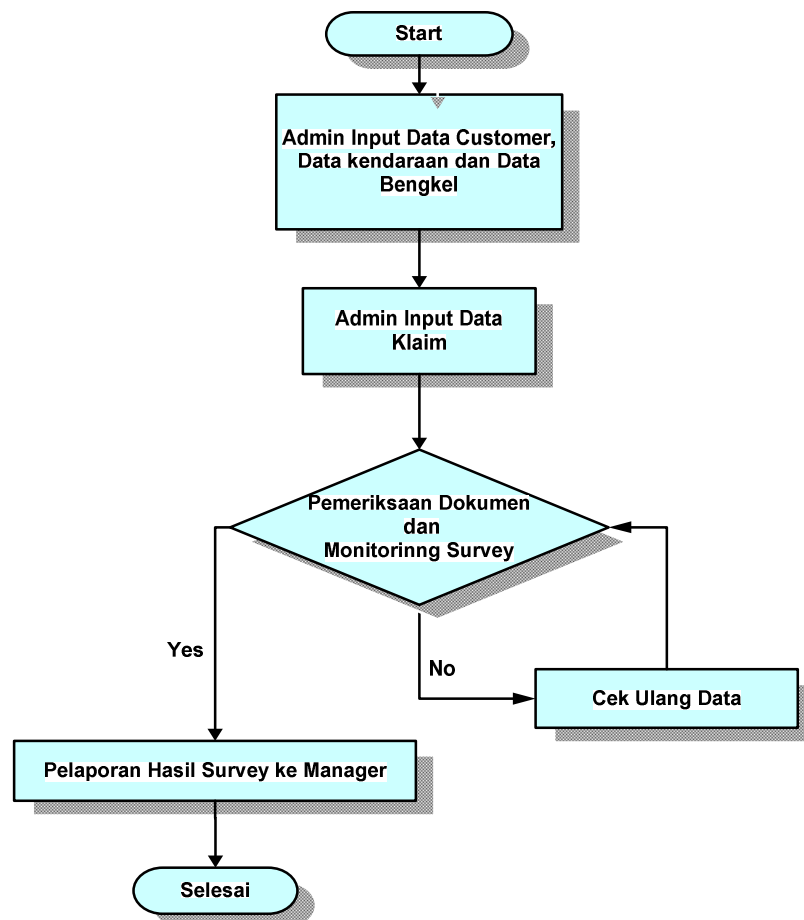
DESAIN DAN PERANCANGAN

3.1 Desain Aplikasi

Aplikasi yang dibuat adalah sebuah aplikasi yang menangani tentang proses monitoring. Aplikasi ini dibuat untuk membantu pihak asuransi untuk membantu pengecekan dan monitoring *survey* pihak bengkel dalam melaksanakan tugasnya.

3.1.1 Flowchart

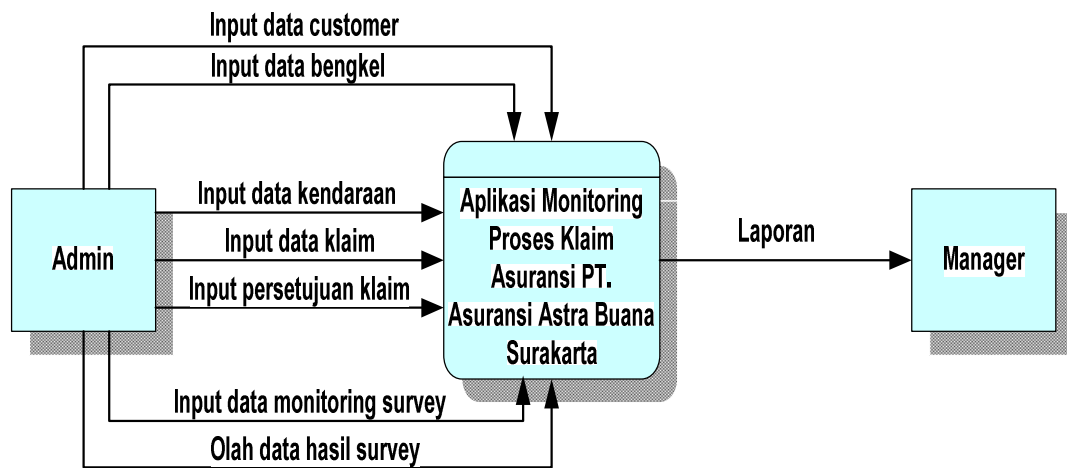
Flowchart proses monitoring klaim asuransi pada PT. Asuransi Astra Buana Surakarta adalah sebagai berikut:



Gambar 3.1 *Flowchart*

3.1.2 Context Diagram (CD)

Context Diagram pada PT. Asuransi Astra Buana Surakarta untuk proses monitoring proses klaim asuransi adalah sebagai berikut:

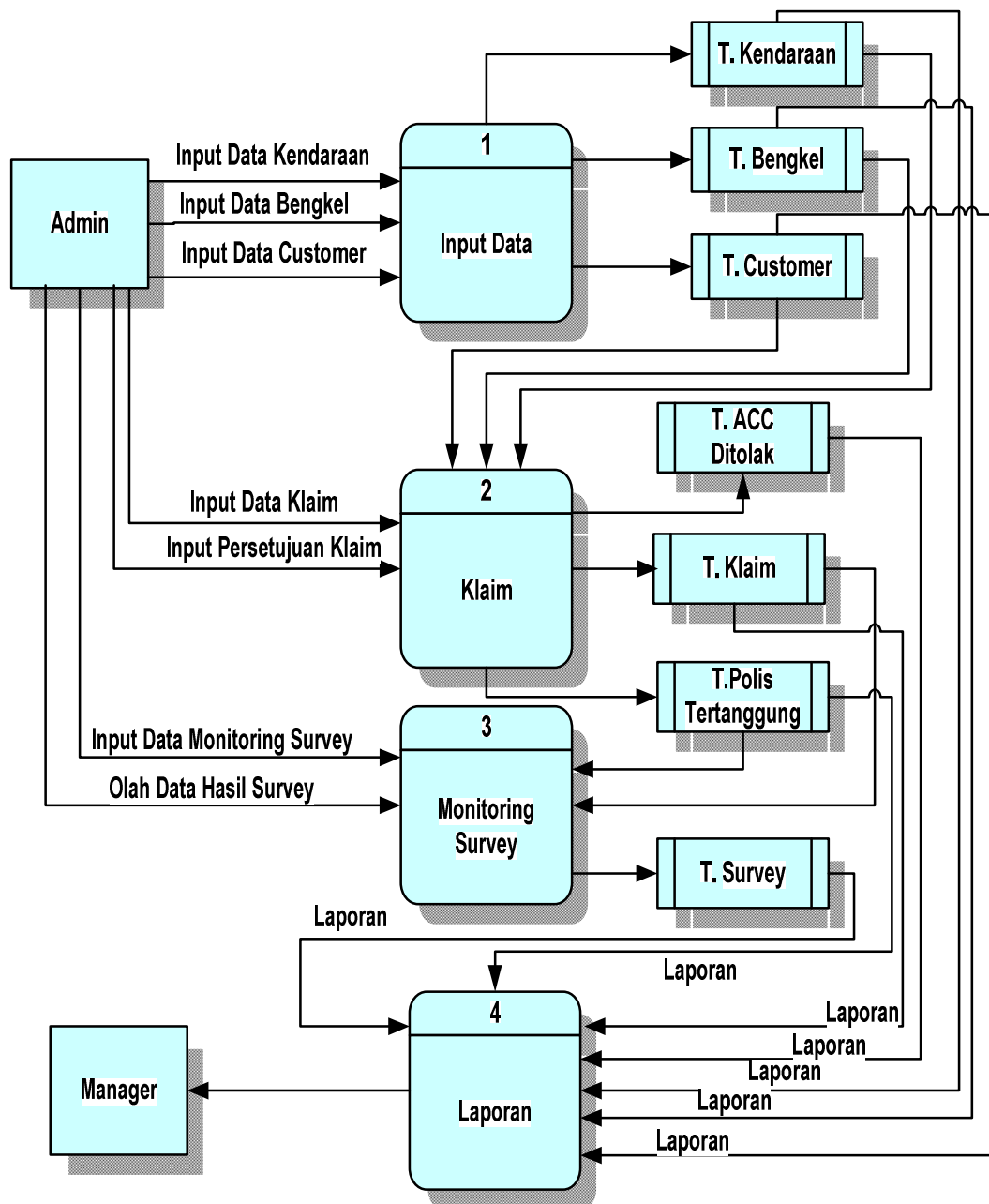


Gambar 3.2 *Context Diagram* (CD)

3.1.3 Data Flow Diagram (DFD)

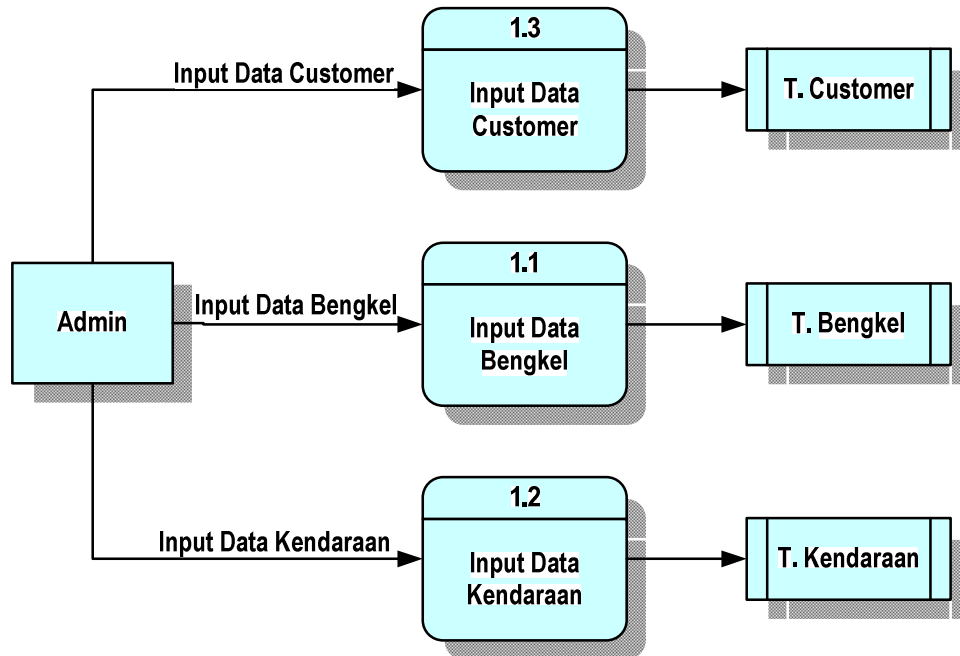
Data Flow Diagram merupakan pengembangan dari *Context Diagram* (CD) *Data Flow Diagram* (DFD) yang terdiri dari DFD level 1 dan DFD level 2 dari pembuatan aplikasi monitoring proses klaim asuransi PT. Asuransi Astra Buana Surakarta.

a. Data Flow Diagram (DFD) Level 1



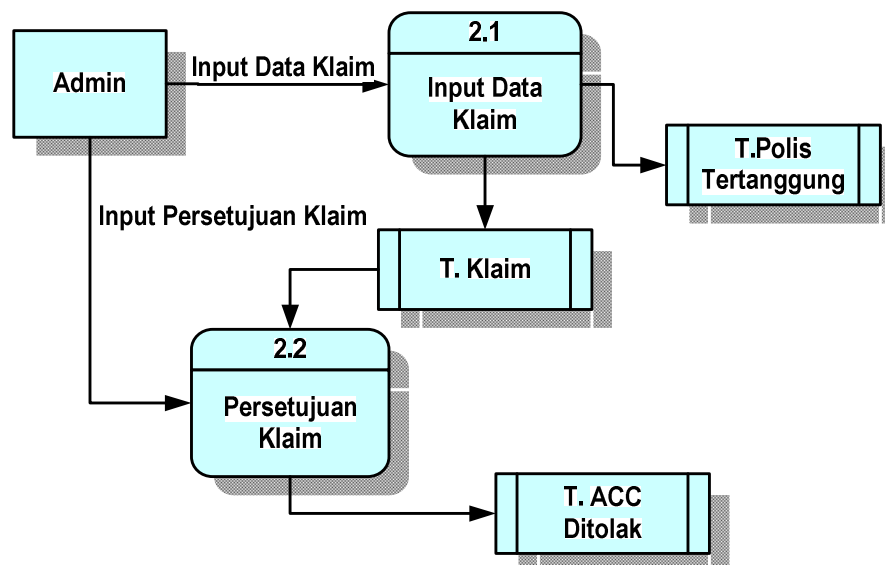
Gambar 3.3 DFD Level 1

b. Data Flow Diagram (DFD) Level 2 Proses 1



Gambar 3.4 (DFD) Level 2 Proses 1

c. Data Flow Diagram (DFD) Level 2 Proses 2



Gambar 3.5 (DFD) Level 1 Proses 2

3.1.4 Data Dictionary

Data dictionary merupakan suatu aplikasi khusus dari jenis kamus-kamus yang digunakan sebagai referensi kehidupan setiap hari. Kamus data merupakan hasil referensi data mengenai data (maksudnya, metadata), suatu data yang disusun oleh penganalisis sistem untuk membimbing mereka selama melakukan analisis dan desain (Kendall & Kendal , 2003).

Kamus data pada pembuatan aplikasi monitoring proses klaim asuransi PT. Asuransi Astra Buana Surakarta sebagai berikut:

1. Data *Customer*

a. Definisi

Informasi mengenai data *customer*.

Data Customer → Customer = @No_Polis + No_Identitas + Nama + Alamat + Pekerjaan + No_Telpon + Ket_1.

b. Atribut

No_Polis*	= { Varchar } 15
No_Identitas	= { Varchar } 25
Nama	= { Varchar } 35
Alamat	= { Varchar } 50
Pekerjaan	= { Varchar } 25
No_Telpon	= { Varchar } 25
Ket_1	= { Varchar } 50

2. Data Kendaraan

a. Definisi

Informasi mengenai data kendaraan *customer*.

Data Kendaraan → Kendaraan = @No_Polis + No_Polisi + No_Rangka + No_Mesin + Merk + Tahun_Rakit + Type + Ket_2.

b. Atribut

No_Polis*	= { Varchar } 15
No_Polisi	= { Varchar } 10
No_Rangka	= { Varchar } 35
No_Mesin	= { Varchar } 35

Merk	= { Varchar } 25
Tahun_Rakit	= { Year } 4
Type	= { Varchar } 25
Ket_2	= { Varchar } 50

3. Data Polis_Tertanggung

a. Definisi

Informasi mengenai data polis asuransi.

Data Polis_Tertanggung → Polis_Tertanggung = @No_Polis + No_Identitas + Nama + Alamat + Pekerjaan + No_Telpn + Ket_1 + No_Polisi + No_Rangka + No_Mesin + Merk + Tahun_Rakit + Type + Ket 2.

b. Atribut

No_Polis*	= { Varchar } 15
No_Identitas	= { Varchar } 16
Nama	= { Varchar } 25
Alamat	= { Varchar } 50
Pekerjaan	= { Varchar } 15
No_Tlpn	= { Varchar } 15
Ket_1	= { Varchar } 50
No_Polisi	= { Varchar } 8
No_Rangka	= { Varchar } 10
No_Mesin	= { Varchar } 10
Merk	= { Varchar } 15
Thn_Rakit	= { Varchar } 4
Type	= { Varchar } 10
Ket_2	= { Varchar } 50

4. Data Survey

a. Definisi

Informasi mengenai data hasil *survey*ajuan klaim *customer*.

Data Survey → Survey = @No_Survey + Waktu_Survey + No_Klaim + No_Polis + Kode_Bengkel + Kerugian + Tim Survey + Telp_Tim + Ket.

b. Atribut

No_Survey* = { Varchar } 25
 Waktu_Survey = { Date }
 No_Klaim = { Varchar } 25
 No_Polis = { Varchar } 25
 Kode_Bengkel = { Varchar } 25
 Kerugian = { Decimal } 10,2
 Tim_Survey = { Varchar } 35
 Telpn_Tim = { Varchar } 25
 Ket = { Varchar } 255

5. Data Bengkel

a. Definisi

Informasi mengenai keterangan tentang rincian bengkel.

Data bengkel → Bengkel = @Kode_Bengkel + Nama + Jenis + Alamat + Ket.

b. Atribut

Kode_Bengkel* = { Varchar } 15
 Nama = { Varchar } 25
 Jenis_Bengkel = { Varchar } 12
 Alamat = { Varchar } 50
 Ket = { Varchar } 50

6. Data Klaim

a. Definisi

Informasi data klaim asuransi.

Data klaim → Klaim = No_Klaim + Waktu + No_Polis + Kerugian + Ket + Survey.

b. Atribut

No_Klaim* = { Varchar } 15

Waktu = {Date}
 No_Polis = {Varchar} 15
 Kerugian = {Varchar} 10,2
 Ket = {Varchar} 50
 Survey = {Integer} 11

7. Data ACC_Ditolak

a. Definisi

Informasi dataajuan klaim yang disetujui.

Data ACC_Ditolak → Acc_diTolak = No_Klaim + No_Polis +
 No_Survey + Tim_Survey + Kode_Bengkel + ACC_Ditolak + Waktu
 + Ket.

b. Atribut

No_Klaim* = {Varchar} 25
 No_Polis = {Varchar} 25
 No_Survey = {Varchar} 25
 Tim_Survey = {Varchar} 35
 Kode_Bengkel = {Varchar} 25
 ACC_Ditolak = {Integer} 11
 Waktu = {Date}
 Ket = {Varchar} 225

3.1.5 Struktur Data

1. Tabel *Customer*

Nama tabel : Customer

Fungsi : Penyimpanan data *customer*.

Tabel 3.1 Tabel *Customer*

Field	Type	Size	Keterangan
No_Polisr*	Varchar	15	Primary Key
No_Identitas	Varchar	25	-
Nama	Varchar	35	-
Alamat	Varchar	50	-
Pekerjaan	Varchar	25	-

No_Telpon	Varchar	25	-
Ket_1	Varchar	50	-

2. Tabel Kendaraan

Nama tabel : Kendaraan

Fungsi : Penyimpanan data kendaraan *customer*.

Tabel 3.2 Tabel Kendaraan

Field	Type	Size	Keterangan
No_Polis*	Varchar	15	Primary Key
No_Polisi	Varchar	10	-
No_Rangka	Varchar	35	-
No_Mesin	Varchar	35	-
Tahun_Rakit	Year	4	-
Type	Varchar	25	-
Ket_2	Varchar	50	-

3. Tabel Polis Tertanggung

Nama tabel : Polis_Tertanggung

Fungsi : Penyimpanan data polis asuransi.

Tabel 3.3 Tabel Polis_Tertanggung

Field	Type	Size	Keterangan
No_Polis*	Varchar	15	Primary Key
No_Identitas	Varchar	16	-
Nama	Varchar	25	-
Alamat	Varchar	50	-
Pekerjaan	Varchar	15	-
No_Tlpn	Varchar	15	-
Ket_1	Varchar	50	-
No_Polisi	Varchar	8	-

No_Rangka	Varchar	10	-
No_Mesin	Varchar	10	-
Merk	Varchar	15	-
Thn_Rakit	Varchar	4	-
Type	Varchar	10	-
Ket_2	Varchar	50	-

4. Tabel *Survey*

Nama tabel : *Survey*

Fungsi : Penyimpanan data hasil *survey*ajuan klaim *customer*.

Tabel 3.4 Tabel *Survey*

Field	Type	Size	Keterangan
No_Survey*	Varchar	25	Primary Key
Waktu_Survey	Date	-	-
No_Klaim	Varchar	25	-
No_Polis	Varchar	25	-
Kode_Bengkel	Varchar	25	-
Kerugian	Decimal	10,2	-
Tim_Survey	Varchar	35	-
Telpn_Tim	Varchar	25	-
Ket	Varchar	225	-

5. Tabel Bengkel

Nama tabel : Bengkel

Fungsi : Penyimpanan keterangan tentang rincian bengkel.

Tabel 3.5 Tabel Bengkel

Field	Type	Size	Keterangan
Kode_Bengkel*	Varchar	15	Primary Key
Nama	Varchar	25	-
Jenis	Varchar	12	-

Alamat	Varchar	50	-
Ket	Varchar	50	-

6. Tabel Klaim

Nama tabel : Klaim

Fungsi : Penyimpanan data klaim asuransi.

Tabel 3.6 Tabel Klaim

Field	Type	Size	Keterangan
No_Klaim*	Varchar	15	Primary Key
Waktu	Date	-	-
No_Polis	Varchar	15	-
Kerugian	Varchar	10,2	-
Ket	Varchar	50	-
Survey	Integer	11	-

7. Tabel ACC di Tolak

Nama Tabel : ACC di Tolak.

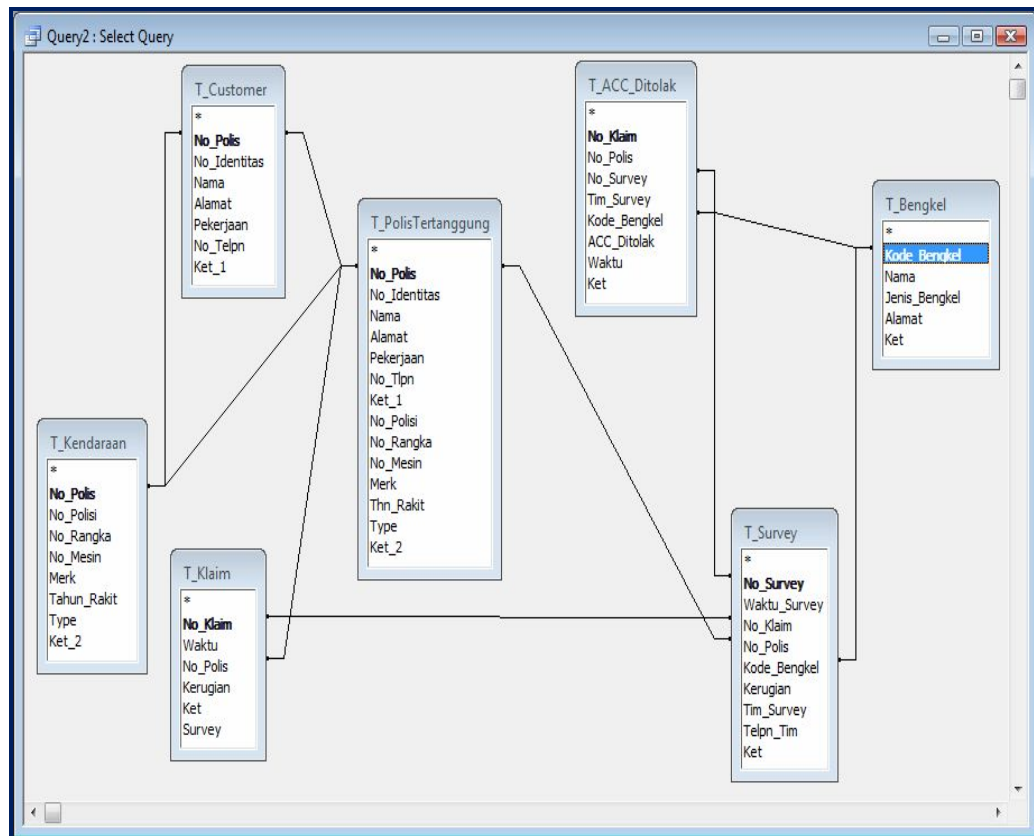
Fungsi : Penyimpanan dataajuan klaim ditolak dan yang disetujui.

Tabel 3.7 Tabel Persetujuan

Field	Type	Size	Keterangan
No_Klaim*	Varchar	25	Primary Key
No_Polis	Varchar	25	-
No_Survey	Varchar	25	-
Tim_Survey	Varchar	35	-
Kode_Bengkel	Varchar	35	-
ACC_Ditolak	Integer	11	-
Waktu	Date	-	-
Ket	Varchar	225	-

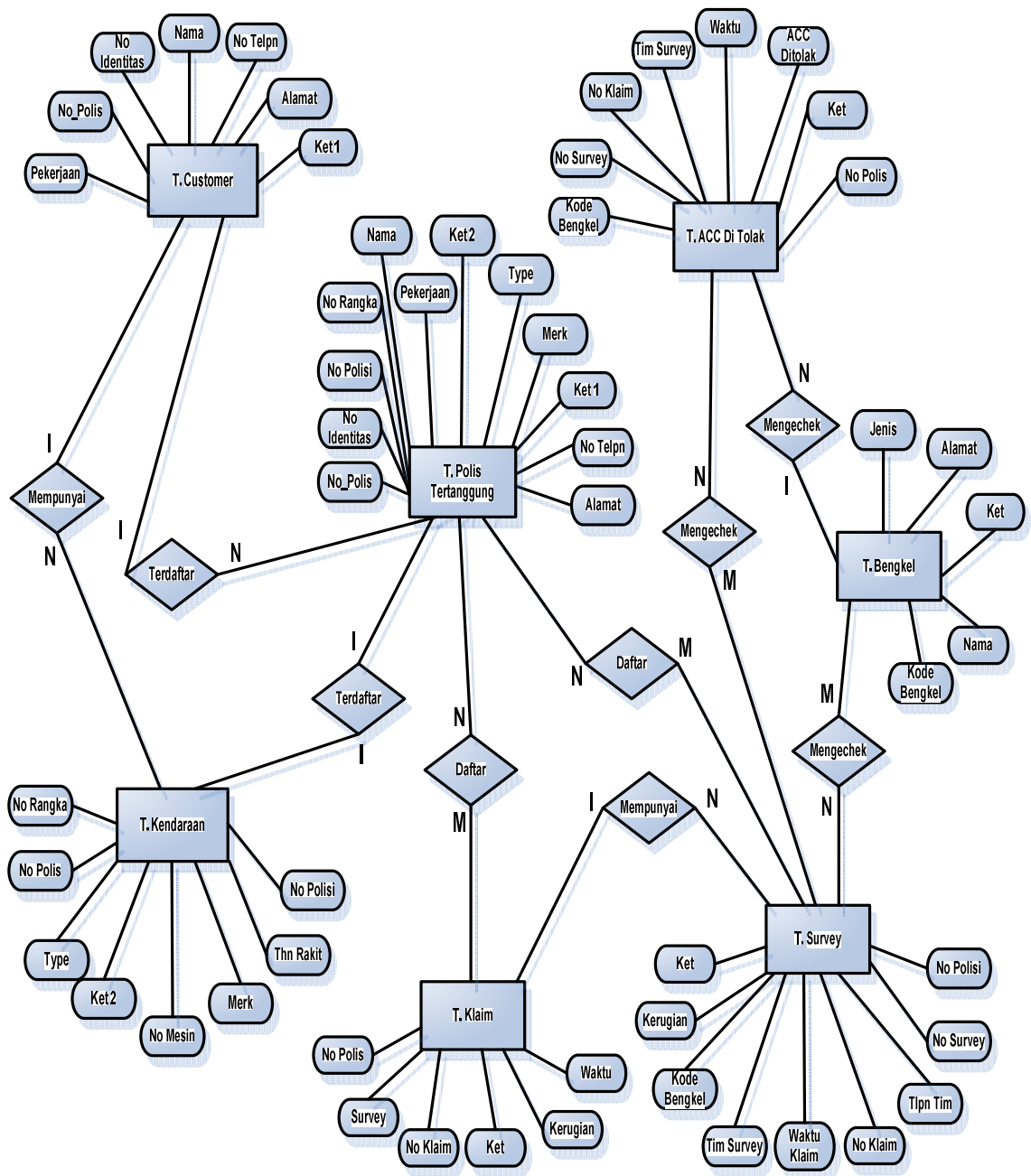
3.1.6 Model Relasional Database (RDBMS)

Berikut adalah gambaran *model relasional database* yang digunakan dalam pembuatan aplikasi monitoring proses klaim asuransi di PT. Asuransi Astra Buana Surakarta.



Gambar 3.6 Model RDBMS

3.1.7 Entity Relationship Diagram (ERD)



Gambar 3.8 *Entity Relationship Diagram (ERD)*

3.2 Perancangan Aplikasi

Dari hasil penelitian ini, dihasilkan sebuah aplikasi program monitoring proses klaim asuransi pada PT. Asuransi Astra Buana Surakarta yang efektif dan

efisien dengan menggunakan *software* aplikasi *Borland Delphi 7.0* dan *database MySQL*.

3.2.1 Kebutuhan *Hardware* dan Kebutuhan *Software*

Untuk mendukung jalannya aplikasi ini, perlu adanya dukungan perangkat komputer yang memadai. Dalam hal ini, diperlukan suatu pertimbangan perangkat komputer yang akan digunakan untuk membangun aplikasi ini secara optimal.

1. Kebutuhan *hardware*

- a. Personal Komputer (PC) yang digunakan untuk dapat membuat aplikasi ini adalah sebagai berikut:

- 1) *Processor* Intel Pentium 1,8 GHz
- 2) *Memory* 480 Mb
- 3) *Hardisk* 80 Gb
- 4) *VGA* 32Mb
- 5) *Monitor*
- 6) *Keyboard*
- 7) *Mouse*

- b. *Printer*

Printer diperlukan dengan pertimbangan kebutuhan dalam pencetakan sebagai hasil laporan.

2. Kebutuhan *software*

- a. Sistem Operasi

Aplikasi ini merupakan aplikasi berbasis *Windows*, sehingga membutuhkan sistem operasi *Windows* serta disarankan menggunakan *Windows XP* atau *Windows Vista*.

- b. *MySQL*

Aplikasi ini membutuhkan *MySQL* sebagai *database*.

- c. *MySQL ODBC*

Aplikasi ini membutuhkan *MySQL ODBC* yang merupakan konektor untuk integrasi dengan *database MySQL*.

BAB IV

IMPLEMENTASI DAN ANALISA

4.1 Implementasi Program

Implementasi program monitoring proses klaim asuransi PT. Asuransi Astra Buana dalam *database* MySQL dibuat dengan menggunakan bahasa pemrograman Borland Delphi 7.0.

4.1.1 Desain *Input*

a. *Form* Utama

Pada *form* utama ini, terdapat beberapa menu yang digunakan untuk menjalankan aplikasi. Menu tersebut didesain khusus untuk memulai jalannya program. Menu-menu tersebut antara lain adalah menu *customer*, menu data bengkel, menu data klaim, menu laporan, menu *survey*, menu *monitoring* dan menu keluar yang berfungsi untuk keluar dari program.



Gambar 4.1 *Form* Utama

b. *Form Data Customer*

Form ini digunakan untuk menginputkan data *customer* yang akan disimpan pada *database*.

No. Polis	No. Identitas	Nama	Alamat
K0989	131456343452	Andreas Adi Sucipto	Semarang
K0213	131990432177	Yanuar Ananta Kusuma	Surabaya
K0812	131895645656	Dinar Khuzuma Wardhani	Malang
K0453	131554231759	Bayu Setya Nugraha	Magelang
K1973	131563321222	Anjar Eka Brastama	Surakarta
K0231	131785664339	Budiman Wicaksono	Yogyakarta

Gambar 4.2 *Form Data Customer*

Keterangan:

Pada saat memulai program, terlebih dahulu menginputkan data data tertanggung dan data kendaraan. Data kendaraan terdiri dari no identitas, no polis, nama, alamat, pekerjaan, no telepon dan keterangan. Sedangkan data kendaraan meliputi no polisi, no rangka, no mesin, merk, tahun perakitan, type dan keterangan. Tombol simpan digunakan untuk menyimpan semua data ke *database* dengan nama tabel data *customer* dan tabel kendaraan. Tombol reset digunakan untuk memulai awal *input* data. Tombol tutup digunakan untuk menutup menu data *customer*. Pilihan menu pada pencarian data ada 2 pilihan yaitu berdasarkan no identitas dan no polis.

c. Form Data Bengkel

Form ini digunakan untuk menginputkan data bengkel yang akan disimpan di *database*.

Kode_Bengkel	Nama	Jenis	Alamat
B45007	Terus Maju Bengkel	NON OTORIS	Jl Ir Sulami 76 Surakarta
B45009	Langgeng Bengkelku	OTORIS	Jl Ahmad Dahlan 79 Surakarta
B45078	Moro Seneng Bengkel	NON OTORIS	Jl Soekarno Hatta 56 Boyolali
B45087	Damai Bengkelku	NON OTORIS	Jl Jayengan 12 Sragen
B45001	Nagano Bengkel	OTORIS	Jl Gajah Mada 34 Sukoharjo
B45011	Armada Masa Depan Bengkel	NON OTORIS	Jl Pemuda 55 Surakarta
B45032	Kali Baru Bengkel	OTORIS	Jl Sumpah Pemuda 67 Surakarta
B45034	Sami Marem Bengkel	NON OTORIS	Jl Anif Rahmad Hakim 32 Sragen
B45065	Harapan Utama Bengkel	NON OTORIS	Jl Mh Thamrin 11 Boyolali
B45043	Kaliaga Bengkel	OTORIS	Jn Slamet Riyadi 89 Sukoharjo

Gambar 4.3 Form Data Bengkel

Keterangan:

Pada menu data bengkel *input* data yang dilakukan meliputi kode bengkel, nama bengkel, alamat jenis bengkel dan keterangan. Tombol simpan digunakan untuk menyimpan data ke *database* dengan nama tabel bengkel. Tombol *reset* digunakan untuk *setting* ulang *input* data, tombol tutup untuk keluar dari program dan tombol tampil semua digunakan untuk menampilkan semua data yang sudah tersimpan pada *database*. Untuk melakukan pencarian data bengkel bisa dilakukan dengan kode bengkel.

d. Form Pengajuan Klaim

Form ini digunakan untuk menginputkan data pengajuan klaim yang diajukan pihak *customer*.

No_Klaim	Waktu	No_Polis	Kerugian	Keterangan
1142	2008/12/02	A890231	1500000	Kaca depan pecah, pintu samping kanan ringsek.
1137	2008/12/02	A890456	550000	Bemper depan retak dan kaca belakang pecah.
1141	2009/05/19	A891456	1500000	Kaca depan pecah, pintu samping kanan ringsek.
1131	2009/05/19	A892564	1500000	Setir bengkok, Kaca depan pecah, pintu samping ka
1141	2009/05/19	A890433	1750000	Lampu seing pecah, beemper hancur.
1145	2009/05/17	A891908	1550000	Bemper depan belakang hancur, setir serong.

Gambar 4.4 *Form Pengajuan Klaim*

Keterangan:

Pada menu pengajuan klaim, untuk no klaim dan waktu harus dilakukan *input* data. Langkah selanjutnya adalah melakukan *input* data no polis, maka secara otomatis semua keterangan data *customer* dan data kendaraan akan muncul karena sebelumnya data sudah tersimpan di *database*. Tombol simpan digunakan untuk menyimpan data ke *database* dengan nama tabel klaim dan tabel polis. Tombol *reset* digunakan untuk *setting* ulang *input* data, tombol tutup untuk keluar dari program dan tombol tampil semua digunakan untuk menampilkan semua data yang sudah tersimpan pada *database*. Untuk melakukan pencarian data pengajuan klaim bisa dengan kode klaim.

e. *Form Klaim ACC/ Gagal ACC*

Form klaim ACC/ gagal ACC ini, digunakan untuk mengecek diterima apa tidak pengajuan klaim yang dilakukan oleh *customer*. Apabila diterima maka klaim acc, tetapi bila tidak diterima maka gagal ACC. Kriteria gagal ACC apabila data tidak sesuai dengan data yang sudah tersimpan pada *database* awal.

Gambar 4.5 *Form Klaim ACC/ Gagal ACC*

Keterangan:

Pada menu klaim ACC/ gagal ACC, data diambil dari data *customer*, data kendaraan dan data bengkel yang sebelumnya sudah tersimpan di *database*. Apabila pengajuan klaim *customer* diterima maka akan lanjut ke proses *survey*. Semua data baik yang ACC maupun gagal ACC, akan tersimpan pada tabel persetujuan.

f. *Form Survey*

Form ini digunakan untuk melakukan *input* data proses *survey* yang dilakukan oleh tim *survey*.

No_Survey	Waktu_Survey	No_Klaim	No_Polis	Kode_Bengkel	Keterangan
001	7/6/2009	1	0789776890	B05	
002	7/6/2009	2	0785489081	B04	
003	7/2/2009	4	0734562210	B02	

Gambar 4.6 *Form Survey*

Keterangan:

Pada menu *survey*, data diambil dari data *customer*, data kendaraan dan data bengkel yang sebelumnya sudah tersimpan di *database*. Selanjutnya juga dilakukan *input* data mengenai no *survey*, waktu *survey*, kerugian, nama tim *survey* dan no telepon. Tombol simpan digunakan untuk menyimpan data ke *database*, tombol *reset* digunakan untuk *setting* ulang *input* data dan tombol tutup untuk keluar dari program. Semua data yang sudah masuk akan disimpan pada tabel *survey*.

g. *Form Monitoring Survey*

Form ini digunakan untuk mengecek kesesuaian antara *customer* yang sudah dilakukan oleh pihak *survey*. Apabila belum diadakan *survey*, maka data tersebut tidak akan muncul.

MONITORING SURVEY	
<div> <div> Data Tertanggung No Klaim: <input type="text"/> Kode Klaim: <input type="text"/> Waktu Klaim: <input type="text"/> No Survey: <input type="text"/> Waktu Survey: <input type="text"/> Besat Kerugian: <input type="text"/> No Polis: <input type="text"/> NoPolis: <input type="text"/> Nama: <input type="text"/> Alamat: <input type="text"/> Pekerjaan: <input type="text"/> No Telepon: <input type="text"/> TIM SURVEY Nama: <input type="text"/> No Telepon/HP: <input type="text"/> </div> <div> <div> Kendaraan Tertanggung No Polisi: <input type="text"/> No Polisi: <input type="text"/> Nomor Rangka: <input type="text"/> No Mesin: <input type="text"/> Merk: <input type="text"/> Type: <input type="text"/> Kode Bengkel: <input type="text"/> Kode Bengkel: <input type="text"/> Nama Bengkel: <input type="text"/> Alamat: <input type="text"/> Jenis Bengkel: <input type="text"/> </div> <div> <div> GASPAROTO Don't worry, Be happy </div> <div> <div> SURVEY TUTUP </div> </div> </div> </div> </div>	

Gambar 4.7 Form Monitoring Survey


Keterangan:

Pada menu monitoring *survey*, langkah pertama program ini adalah pihak *customer* mendaftarkan pengajuan klaim, setelah itu *customer* tersebut bisa mendapatkan no klaim. Selanjutnya bila pengajuan klaim disetujui dan sudah dilakukan perbaikan maka apabila dilakukan monitoring *survey*, data *customer* akan muncul tanpa ada peringatan. Apabila data *customer* tersebut belum terdaftar sebagai pengajuan klaim, maka data tidak muncul dan harus dilakukan proses *survey* dahulu.

4.1.2 Desain Output

a. Laporan Data *Customer*

Laporan ini berisi tentang data pribadi para *customer* yang ikut menjadi mitra kerja dengan pihak asuransi. Untuk mendapatkan data tersebut, sebelumnya harus melakukan *input* data yang selanjutnya data akan disimpan di *database*.




DATA CUSTOMER		
No Polis	K089	Keterangan :
No Identitas	131423114688	Customer
Nama	Andreastomo Budi Santoso	
Alamat	Jl Kaliurang No 12 Yogyakarta	
Pekerjaan	Dokter	
No Telepon	081226756443	

Gambar 4.8 Laporan Data *Customer*

b. Laporan Data Bengkel

Pada laporan data bengkel, data diambil dari *database* yang sebelumnya sudah dilakukan *input* data. Output pada data bengkel diklasifikasikan antara jenis bengkel otoris dan non otoris.



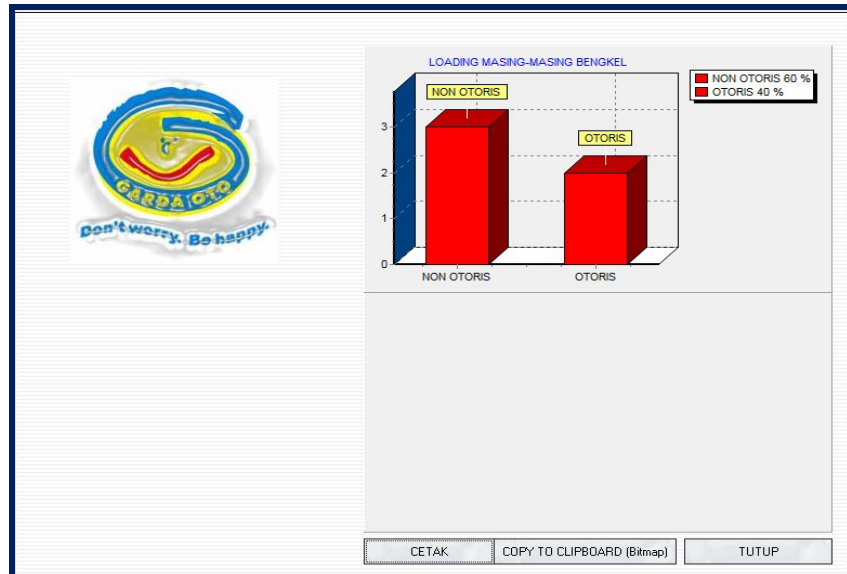
DATA BENGKEL				
Kode Bengkel	Nama Bengkel	Jenis	Alamat	Keterangan
B04	BAM	OTORIS	Jl Rajiman Shaleh 88 Surak	Bengkel Otoris
B02	ISO	OTORIS	Jl Krakatau 87 Sumber Sur	Bengkel Otoris
B01	DSO	OTORIS	Kartosura	Bengkel Otoris

Gambar 4.9 Laporan Data Bengkel

c. Laporan Grafik *Loading* Bengkel

Laporan ini berisi tentang prosentase jumlah bengkel *otoris* dan *non otoris*. Bengkel *otoris* merupakan bengkel yang hanya melayani konsumen yang memiliki mobil dari produksi *astra*. Sedangkan bengkel *non otoris* merupakan bengkel umum yang


menjadi mitra kerja dengan pihak asuransi yang melayani konsumen yang memiliki mobil selain produksi astra.



Gambar 4.10 Laporan *Loading Bengkel*

d. Laporan Klaim ACC dan Gagal ACC

Pada laporan klaim ACC dan gagal ACC merupakan laporan dari pihak asuransi untuk menyetujui pengajuan klaim dari pihak *customer*. Laporan ini bertujuan untuk mengecek keberadaan *customer* apakah telah tergabung menjadi mitra kerja dengan pihak asuransi.

		Data Pengajuan Klaim <u>ACC</u>			
No Klaim	No Polis	Tim Survey	Bengkel	Waktu	Keterangan
1	0789776890	Gunawan	B05	7/7/2009	ACC
2	0765489081	Sarjono Sumartono	B04	6/29/2009	ACC

Gambar 4.11 Laporan Klaim ACC dan Gagal ACC

4.2 Analisa Program

Subbab ini membahas tentang analisis hasil yang telah diimplementasikan pada program yang sebenarnya. Program aplikasi monitoring menggunakan *database* MySQL dan bahasa pemrograman Borland Delphi 7.0

4.2.1 Keunggulan Aplikasi Program

Aplikasi ini mempermudah pihak asuransi untuk melakukan proses monitoring data dan membantu dalam membuat laporan.

Keunggulan aplikasi ini adalah lebih efektif dan efisien dibandingkan dengan sebelumnya yang hanya menggunakan *Microsoft Excel*.

4.2.2 Kelemahan Aplikasi Program

Aplikasi monitoring proses klaim asuransi ini, memiliki kelemahan yaitu masih bersifat *stand alone* belum secara *client server online* sehingga proses akses dalam memonitoring data tidak dapat dilakukan dari luar instansi.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian ini, maka penulis dapat mengambil kesimpulan bahwa aplikasi monitoring proses klaim asuransi PT. Asuransi Astra Buana di buat dengan bahasa pemrograman *Borland Delphi 7.0* dan *Database MySQL* dengan konektor *ODBC (Open Database Conectivity) 5.15- win32*.

5.2 Saran

Berdasarkan kesimpulan dari hasil penelitian ini, dapat diambil saran sebagai berikut:

1. Aplikasi ini belum memiliki *user name* dan *password* sehingga data dapat dengan mudah dan bebas diakses oleh pihak yang tidak berkepentingan, akan lebih baik jika dikembangkan lagi dengan menambahkan *user name* dan *password*.
2. Aplikasi ini masih bersifat *stand alone* akan lebih baik diterapkan secara *client server online* sehingga mempermudah dalam mengakses untuk memonitoring dari luar.

DAFTAR PUSTAKA

- Edhy Sutanta, 2004. *Sistem Basis Data*, Graha Ilmu, Yogyakarta. hal. 2-7
- Fariad Irmansyah, 2003, *Pengantar Database*, www.ilmukomputer.com, 14 Maret 2009.
- Fathansyah, 2004, *Sistem Basis Data Lanjutan Buku Basis Data*, Informatika Bandung, Bandung. hal. 12-14
- Hariyanto Kristanto, 1994, *Konsep & Perancangan Database*, Andi Offset, Yogyakarta. hal. 1-8
- Jogiyanto H. M, 2001, *Analisa dan Desain Sistem Informasi*, Andi Offset, Yogyakarta. Hal 203-215
- Kadir, Abdul, 2001, *Konsep & Tuntunan Praktis Basis Data*, Andi Offset, Yogyakarta. hal. 6-11
- Kadir, Abdul, 2003, *Dasar Aplikasi Database MySQL Delphi*, Andi Offset, Yogyakarta. hal. 2-4 dan hal. 35-43
- Kendall & Kendall, 2003, *Analisis Perancangan Sitem Edisi Kelima*, PT. Indeks (Kelompok Gramedia), Jakarta. hal. 333-334
- Krida, 2009, Pengertian Basis Data, krida85.wordpress.com, 15 Maret 2009.
- Supriadi, Muhammad, 2007, *Pemrograman IC PPI 8255 Menggunakan Delphi*, Andi Offset, Yogyakarta. hal. 1-2